



(11) Publication number : **0 532 228 A2**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number : **92308000.6**

(51) Int. Cl.⁵ : **H04L 9/32, H04L 9/06**

(22) Date of filing : **03.09.92**

(30) Priority : **13.09.91 US 759309**

(43) Date of publication of application :
17.03.93 Bulletin 93/11

(84) Designated Contracting States :
DE FR GB SE

(71) Applicant : **AMERICAN TELEPHONE AND
TELEGRAPH COMPANY**
32 Avenue of the Americas
New York, NY 10013-2412 (US)

(72) Inventor : **Reads III, James Alexander**
127 Southgate Road
New Providence, New Jersey 07974 (US)

(74) Representative : **Buckley, Christopher Simon**
Thirsk et al
AT & T (UK) Ltd. 5 Morningside Road
Woodford Green Essex IG8 OTU (GB)

(54) **A cryptosystem for cellular telephony.**

(57) A relatively secure, self-inverting, symmetric key cryptosystem designed for efficient implementation on an 8-bit microcomputer. The cryptosystem is especially well suited use in cellular telephony. The method of encryption is comprised of three stages : 1) an autokeyed encryption, 2) the use of a one-time pad encryption where the key is derived from a portion of the message as encrypted by the first stage, and 3) a second autokeyed decryption that is the inverse of the first.

EP 0 532 228 A2

Background of the Invention

Field of the Invention

This invention relates to cryptology and more particularly to a cryptosystem for ensuring the privacy of communications in the context of cellular telephony.

Description of the Prior Art

In conventional telephony each telephone set (fax unit, modem, etc) is physically connected to a unique port on a switch at a local central office. The connection is through a dedicated wire, or through a designated channel on a dedicated wire. The wire connection is installed by the service provider (who, typically, is the common carrier) and, therefore, the service provider can be reasonably sure that transmission on the channel arrives from the subscriber. By comparison, authentication of a subscriber in wireless telephony is less certain.

Under the current cellular telephony arrangement in the United States, when a cellular telephone subscriber places a call, his or her cellular telephone indicates to the service provider the identity of the caller for billing purposes. This information is not encrypted. If an interloper eavesdrops at the right time, he or she can obtain the subscriber's identification information. This includes the subscriber's phone number and the electronic serial number (ESN) of the subscriber's equipment. Thereafter, the interloper can program his or her cellular telephone to impersonate that *bona fide* subscriber to fraudulently obtain services. Alternately, an interloper can inject himself into an established connection, overpower the customer's cellular telephone transmitter by transmitting more power, and redirect the call to his or her purposes by sending certain control codes to the service provider. Basically, such piracy will succeed because the service provider has no mechanism for independently authenticating the identity of the caller at the time the connection is established and/or while the connection is active.

Technology is available to permit an eavesdropper to automatically scan all of the cellular frequencies in a given cell for such identification information. Consequently, piracy of cellular telephone services is rampant. Also, the lack of enciphering of the speech signals lays bare to eavesdroppers the content of conversations. In short, there is a clear and present need for effective security measures in the cellular telephony art, and that suggests the use of cryptology for the purposes of ensuring authentication and privacy.

Several standard cryptographic methods exist for solving this general sort of authentication problem that exist in cellular telephony, but they turn out to

have practical problems. First, a classical challenge/response protocol may be used, based on a private key cryptographic algorithm. In this approach, a subscriber's mobile station is issued with a secret key which also known by the home system. When a serving system wishes to authenticate a subscriber, it applies to the home system for a challenge and a response to use with the given subscriber. The home system composes a random challenge and applies a one-way function to the challenge concatenated with the subscribers key to obtain the corresponding response. The challenge and response are supplied to the serving system, which issues the challenge to the mobile station. The mobile station in turn replies with the response, which it calculates from the challenge and from its stored secret key. The serving system compares the responses supplied by the home system and by the mobile station, and if they match, the mobile station is deemed authentic.

The problem with this approach is that often the serving system is unable to contact the home system quickly enough to allow authentication of a call setup, or that the database software on the home system is unable to look up the subscriber's secret key and compose the challenge/response pair quickly enough. Network or software delays of a second or two would add that much dead time till the subscriber hears a dial tone after picking up the handset when placing a call, and longer delays (given the control networks and switching apparatus currently used by cellular providers) would be common. In the present milieu, such delays are unacceptable.

Public key cryptography provides another standard class of ways for solving authentication problems. Generally speaking, each mobile station would be provided with a "public key certificate" of identity, signed by the public key of the service provider, stating that the mobile station is a legitimate customer of the service provider. In addition, each mobile would also be given secret data (private keys) which it can use, together with the certificate, to prove to third parties (such as the serving system) that it is a legitimate customer.

For example, service provider could have a pair of RSA keys, (F,G), with F private and G public. The service provider could supply each mobile with its own pair (D,E) of RSA keys, together with F(E) (the encryption of the mobile's public key E using the provider's private key F). Then a mobile asserts its identity by sending (E,F(E)) to the serving system. The serving system applies G to F(E) to obtain E. The serving system generates a challenge X, encrypts it with the mobile's public key E to obtain E(X) which it sends to the mobile. The mobile applies its private key D to E(X) to obtain X, which it sends back to the server in the clear as a response.

Although some variations on this theme involve less computation or data transmission than others, no

public key authentication scheme yet exists which is efficiently executable in less than a second's time on the sort of hardware currently used in cellular telephones. Even though network connectivity between the serving and home systems is not needed at the moment of authentication, as it is in the classical approach, the same time constraints which rule out the classical approach also rule out the public key approach.

Summary of the Invention

In accordance with the invention, plaintext messages of variable length can be encrypted and decrypted easily on an 8-bit microcomputer. The invention is a relatively secure, self-inverting, symmetric key cryptosystem that is comprised of three stages. The first stage is an autokeyed encryption on the plaintext. The second stage is a self-inverting cipher where the encryption key is derived from a portion of the message as encrypted by the first stage. The third stage is a second autokeyed decryption that corresponds to the autokeyed encryption of the first stage.

Brief Description of the Drawing

FIG. 1 illustrates an arrangement of network providers and cellular radio providers interconnected for service to both stationary and mobile telephones and the like;

FIG. 2 depicts the process for directing the creation of a shared secret data field and the verification of same;

FIG. 3 shows the elements that are concatenated and hashed to create the shared secret data;

FIG. 4 shows the elements that are concatenated and hashed to create the verification sequence;

FIG. 5 shows the elements that are concatenated and hashed to create the registration sequence when the mobile unit goes on the air;

FIG. 6 shows the elements that are concatenated and hashed to create the call initiation sequence;

FIG. 7 depicts the speech encryption and decryption process in a mobile unit;

FIG. 8 shows the elements that are concatenated and hashed to create the re-authentication sequence;

FIG. 9 illustrates the three stage process for encrypting and decrypting selected control and data messages; and

FIG. 10 presents a block diagram of a mobile unit's hardware.

Detailed Description

In a mobile cellular telephone arrangement there are many mobile telephones, a much smaller number

of cellular radio providers (with each provider having one or more base stations) and one or more switching network providers (common carriers). The cellular radio providers and the common carriers combine to allow a cellular telephone subscriber to communicate with both cellular and noncellular telephone subscriber. This arrangement is depicted diagrammatically in FIG. 1, where common carrier I and common carrier II combine to form a switching network comprising switches 10-14. Stationary units 20 and 21 are connected to switch 10, mobile units 22 and 23 are free to roam, and base stations 30-40 are connected to switches 10-14. Base stations 30-34 belong to provider 1, base stations 35 and 36 belong to provider 2, base station 37 belongs to provider 4, and base stations 38-40 belong to provider 3. For purposes of this disclosure, a base station is synonymous with a cell wherein one or more transmitters are found. A collection of cells makes up a cellular geographic service area (CGSA) such as, for example, base stations 30, 31, and 32 in FIG. 1.

Each mobile unit has an electronic serial number (ESN) that is unique to that unit. The ESN number is installed in the unit by the manufacturer, at the time the unit is built (for example, in a read-only-memory), and it is unalterable. It is accessible, however.

When a customer desires to establish a service account for a mobile unit that the customer owns or leases, the service provider assigns to the customer a phone number (MIN1 designation), an area code designation (MIN2 designation) and a "secret" (A-key). The MIN1 and MIN2 designations are associated with a given CGSA of the provider, and all base stations in the FIG. 1 arrangement can identify the CGSA to which a particular MIN2 and MIN1 pair belongs. The A-key is known only to the to the customer's equipment and to provider's CGSA processor (not explicitly shown in FIG. 1. The CGSA processor maintains the unit's ESN, A-key, MIN1 and MIN2 designations and whatever other information the service provider may wish to have.

With the MIN1 designation and the A-key installed, the customer's unit is initialized for service when the CGSA processor sends to the mobile unit a special random sequence (RANDSSD), and a directive to create a "shared secret data" (SSD) field. (The CGSA sends the RANDSSD and the SSD field generation directive through the base station of the cell where the mobile unit is present.) Creation of the SSD field follows the protocol described in FIG. 2.

As an aside, in the FIG. 1 arrangement each base station broadcasts information to all units within its cell on some preassigned frequency channel (broadcast band). In addition, it maintains two way communications with each mobile unit over a mutually agreed, (temporarily) dedicated, channel. The manner by which the base station and the mobile unit agree on the communications channel is unimportant

to this invention, and hence it is not described in detail herein. One approach may be, for example, for the mobile unit to scan all channels and select an empty one. It would then send to the base station its MIN2 and MIN1 designations (either in plaintext form or enciphered with a public key), permitting the base station to initiate an authentication process. Once authenticated communication is established, if necessary, the base station can direct the mobile station to switch to another channel.

As described in greater detail hereinafter, in the course of establishing and maintaining a call on a mobile telephony system of this invention, an authentication process may be carried out a number of times throughout the conversation. Therefore, the authentication process employed should be relatively secure and simple to implement. To simplify the design and lower the implementation cost, both the mobile unit and the base station should use the same process.

Many authentication processes use a hashing function, or a one-way function, to implement the processes. A hashing function performs a many-to-one mapping which converts a "secret" to a signature. The following describes one hashing function that is simple, fast, effective, and flexible. It is quite suitable for the authentication processes of this invention but, of course, other hashing functions can be used.

The Jumble Process

the Jumble process can create a "signature" of a block of d "secret" data words $b(i)$, with the aid of a k -word key $x(j)$, where d and k are integers. The "signature" creation process is carried out on one data word at a time. For purposes of this description, the words on which the Jumble process operates are 8 bits long (providing a range from 0 to 256), but any other word size can be employed. The "secret" data block length is incorporated in the saw tooth function

$$s_d(t) = t \text{ for } 0 \leq t \leq d-1$$

$$s_d(t) = 2d-2-t \text{ for } d \leq t \leq 2d-3, \text{ and}$$

$$s_d(t) = s_d(t+2d-2) \text{ for all } t.$$

This function is used in the following process where, starting with $z=0$ and $i=0$, for successively increasing integer values of i in the range $0 \leq i \leq 6d-5$,

a) $b(s_d(i))$ is updated by:

$$b(s_d(i)) = b(s_d(i)) + x(i_k) + \text{SBOX}(z) \bmod 256$$

where i_k is i modulo k , $\text{SBOX}(z) = y + [y/2048] \bmod 256$,

$y = (z \oplus 16)(z + 111)(z)$, $[y/2048]$ is the integer portion of y divided by 2048, and \oplus represents the bit-wise Exclusive-OR function; and

b) z is updated with: $z = z + b(s_d(i)) \bmod 256$.

It may be appreciated that in the process just described there is no real distinction between the data and the key. Therefore, any string that is used for authentication can have a portion thereof used as a key for the above process. Conversely, the data words

concatenated with the key can be considered to be the "authentication string." It may also be noted that each word $b(i)$, where $0 \leq i < d-1$ is hashed individually, one at a time, which makes the hashing "in place". No additional buffers are needed for the hashing process *per se*.

The process just described can be easily carried out with a very basic conventional processor, since the only operations required are: shifting (to perform the division by 2048), truncation (to perform the $[]$ function and the mod 256 function), addition, multiplication, and bit-wise Exclusive-OR functions.

Returning to the SSD field initialization process of FIG. 2, when a RANDSSD sequence and the directive to create a new SSD field (arrow 100 in FIG. 2) are received by the mobile station, a new SSD field is generated in accordance with FIG. 3. The mobile unit concatenates the ESN designation, the A-key, and the RANDSSD sequence to form an authentication string. The authentication string is applied to Jumble block 101 (described above) which outputs the SSD field. The SSD field comprises two subfields: the SSD-A subfield which is used to support authentication procedures, and the SSD-B subfield which is used to support voice privacy procedures and encryption of some signaling messages (described below). It may be noted that a larger number of SSD subfields can be created. If one needs a larger total number of bits, one needs only to start with a larger number of data bits. As will be appreciated from the disclosure below, that is not a challenging requirement.

The home CGSA processor knows the ESN and the A-key of the mobile unit to which the received MIN2 and MIN1 designations were assigned. It also knows the RANDSSD sequence that it sent. Therefore, the home CGSA processor is in position to duplicate the SSD field creation process of the mobile unit. By concatenating the RANDSSD signal with the ESN designation and the A-key, and with the above-described Jumble process, the CGSA processor creates a new SSD field and partitions it into SSD-A and SSD-B subfields. However, the SSD field created in the home CGSA processor must be verified.

In accordance with FIG. 2, verification of the created SSD field is initiated by the mobile unit. The mobile unit generates a random challenge sequence (RANDBS sequence) in block 102 and sends it to the home CGSA processor through the serving base station. In accordance with FIG. 4, the home CGSA processor concatenates the challenge RANDBS sequence, the ESN of the mobile unit, the MIN1 designation of the mobile unit, and the newly created SSD-A to form an authentication string which is applied to the Jumble process. In this instance, the Jumble process creates a hashed authentication signal AUTHBS which is sent to the mobile station. The mobile station also combines the RANDBS sequence, its

ESN designation, its MIN1 designation and the newly created SSD-A to form an authentication string that is applied to the Jumble process. The mobile station compares the result of its Jumble process to the hashed authentication signal (AUTHBS) received from the home CGSA processor. If the comparison step (block 104) indicates a match, the mobile station sends a confirmation message to the home CGSA processor indicating the success of the update in the SSD field. Otherwise, the mobile station reports on the failure of the match comparison.

Having initialized the mobile station, the SSD field remains in force until the home CGSA processor directs the creation of a new SSD field. That can occur, for example, if there is reason to believe that the SSD field has been compromised. At such a time, the home CGSA processor sends another RANDSSD sequence to the mobile unit, and a directive to create a new SSD field.

As mentioned above, in cellular telephony each base station broadcasts various informational signals for the benefit of all of the mobile units in its cell. In accordance with FIG. 1 management, one of the signals broadcast by the base station is a random or pseudorandom sequence (RAND sequence). The RAND sequence is used by various authentication processes to randomize the signals that are created and sent by the mobile units. Of course, the RAND sequence must be changed periodically to prevent record/playback attacks. One approach for selecting the latency period of a RAND signal is to make it smaller than the expected duration of an average call. Consequently, a mobile unit, in general, is caused to use different RAND signals on successive calls.

In accordance with one aspect of this invention, as soon as the mobile unit enters a cell it registers itself with the base unit so that it can be authenticated, it can initiate calls, and the base station can direct calls to the mobile unit. When the mobile unit registers with a serving base station, it sends to the serving base station its MIN1 and MIN2 designations and its ESN sequence. Of course, an authentication process is carried out in the course of registering, and that process is depicted in FIG. 5. According to FIG. 5, the mobile unit receives the broadcast RAND sequence; concatenates the RAND sequence, the ESN sequence, the MIN1 designation and the SSD-A subfield to form an authentication string; and applies the authentication string to the Jumble process. The hashed authentication string at the output of the Jumble process is sent to the serving base station together with the ESN sequence.

In some embodiments, all or part of the RAND sequence used by the mobile unit is also sent to the serving base station, because the possibility exists that the RAND value has changed by the time the hashed authentication string reaches the base station.

On the base station side, the serving base station knows the RAND sequence because the base station created it. The base station also knows the ESN and the MIN2 and MIN1 designations that the mobile unit identified itself with. But, on first registration, the serving base station does *not* know the SSD field of the mobile unit. It does know, however, the mobile unit's home CGSA processor (from the MIN1 and MIN2 designations) so the authentication proceeds as follows. The serving base station sends to the home CGSA processor the MIN1 designation, the ESN sequence, the hashed authentication string that the mobile unit transmitted, and the RAND sequence that the mobile unit used to create the hashed authentication string. From the mobile unit's MIN1 designation and ESN sequence the home CGSA processor knows the mobile unit's SSD-A subfield, so it proceeds to create an authentication string as described above and applies it to the Jumble process. If the hashed authentication string created by the home CGSA processor matches the hashed authentication string created in the mobile unit and supplied by the serving base station, then verification is deemed successful. In such a case, the home CGSA processor supplies the serving base station with the unit's SSD field. As an aside, to keep the ESN designation and the SSD field secure, the communication between the base stations and the CGSA processor is carried in encrypted form.

Once the mobile unit has been authenticated at the serving base station (via the above-described process) the serving base station possesses the ESN and the SSD field of the mobile unit, and subsequent authentication processes in that cell can proceed in the serving base station without reference to the home CGSA processor -- except one. Whenever, for any reason, it is desirable to alter the SSD field, communication is effectively between the home CGSA processor and the mobile unit; and the serving base station serves only as a conduit for this communication. That is, the home CGSA processor creates a RANDSSD sequence and alters the SSD field based on that RANDSSD sequence, the home CGSA processor supplies the serving base station with the RANDSSD sequence and the newly created SSD field, the serving base station directs the mobile unit to alter its SSD field and provides the mobile unit with the RANDSSD sequence, the mobile unit alters the SSD field and sends a challenge to the serving base station, the serving base station creates the AUTHBS string (described above) and sends it to the mobile unit, and the mobile unit verifies the AUTHBS string and informs the serving base station that both the mobile unit and the serving base station have the same SSD fields.

Having been registered by the serving base station, the mobile unit can initiate calls, in accordance with FIG. 6. The call initiation sequence concatenates

signals RAND, ESN, SSD-A and at least some of the called party's identification (phone) number (MIN3 in FIG. 6). The concatenated signals are applied to the Jumble process to develop a hashed authentication sequence that can be verified by the serving base station. Of course, to permit verification at the base station, the called party's identification number must also be transmitted in a manner that can be received by the base station (and, as before, perhaps a portion of the RAND signal), i.e., in plaintext. Once the authentication sequence is verified, the base station can process the call and make the connection to the called party.

The protocol for connecting to a mobile unit when it is a "called party", follows the registration protocol of FIG. 5. That is, the serving base station requests the called mobile station to send an authentication sequence created from the RAND sequence, ESN designation, MIN1 designation and SSD-A subfield. When authentication occurs, a path is set up between the base station and the called party mobile unit, for the latter to receive data originating from, and send data to, the mobile unit (or stationary unit) that originated the call.

It should be noted that all of the authentications described above are effective only (in the sense of being verified) with respect to the authenticated packets, or strings, themselves. To enhance security at other times, three different additional security measures can be employed. They are speech encryption, occasional re-authentication, and control message encryption.

Speech Encryption

The speech signal is encrypted by first converting it to digital form. This can be accomplished in any number of conventional ways, with or without compression, and with or without error correction codes. The bits of the digital signals are divided into successive groups of K bits and each of the groups is encrypted. More specifically, in both the mobile unit and the base station the RAND sequence, the ESN and MIN1 designations, and the SSD-B subfield are concatenated and applied to the Jumble process. The Jumble process produces 2K bits and those bits are divided into groups A and B of K bits each. In the mobile unit group A is used for encrypting outgoing speech, and group B is used for decrypting incoming speech. Conversely in the base station, group A is used for decrypting incoming speech and group B is used for encrypting outgoing speech. FIG. 7 depicts the speech encryption and decryption process.

Re-authentication

At the base station's pleasure, a re-authentication process is initiated to confirm that the mobile unit

which the base station believes is active, is, in fact, the mobile unit that was authorized to be active. This is accomplished by the base station requesting the mobile unit to send a hashed authentication sequence in accordance with FIG. 8. With each such request, the base station sends a special (RANDU) sequence. The mobile unit creates the hashed authentication sequence by concatenating the RANDU sequence, the area code MIN2 designation of the mobile unit, the ESN designation, the MIN1 designation and the SSD-A designation. The concatenated string is applied to the Jumble process, and the resulting hashed authentication string is sent to the base station. The base station, at this point, is in a position to verify that the hashed authentication string is valid.

Control Message Cryptosystem

The third security measure deals with ensuring the privacy of control messages. In the course of an established call, various circumstances may arise that call for the transmission of control messages. In some situations, the control messages can significantly and adversely affect either the mobile station that originated the call or the base station. For that reason, it is desirable to encipher (reasonably well) some types of control messages sent while the conversation is in progress. Alternately, selected fields of selected message types may be encrypted. This includes "data" control messages such as credit card numbers, and call redefining control messages. This is accomplished with the Control Message Cryptosystem.

The Control Message Cryptosystem (CMC) is a symmetric key cryptosystem that has the following properties:

- 1) it is relatively secure,
- 2) it runs efficiently on an eight-bit computer, and
- 3) it is self-inverting (i.e., involutory).

The cryptographic key for CMC is an array, TBOX[z], of 256 bytes which is derived from a "secret" (e.g., SSD-B subfield) as follows: for each z in the range $0 \leq z < 256$, set TBOX[z]=z, and apply the array TBOX[z] and the secret (SSD-B) to the Jumble process.

This is essentially what is depicted in elements 301, 302 and 303 in FIG. 7 (except that the number of bytes in FIG. 7 is 2K rather than 256).

Once the key is derived, CMC can be used to encrypt and decrypt control messages. Alternately, the key can be derived "on-the-fly" each time the key is used. CMC has the capability to encipher variable length messages of two or more bytes. CMC's operation is self-inverting, reciprocal, or involutory. That is, precisely the same operations are applied to the ciphertext to yield plaintext as are applied to plaintext to yield ciphertext. An involutory function is a function

which is its own inverse (e.g., $x = \frac{1}{x'}$, $x = T(T(x'))$). Thus, a two-fold application of the CMC operations would leave the buffer contents unchanged.

In the description that follows it is assumed that for the encryption process (and the decryption process) the plaintext (or the ciphertext) resides in a data buffer and that CMC operates on the contents of that data buffer such that the final contents of the data buffer constitute the ciphertext (or plaintext). That means that elements 502 and 504 in FIG. 9 can be one and the same register.

CMC is comprised of three successive stages, each of which alters each byte string in the data buffer. Note that both CMC, as a whole, and the second constituent stage of CMC are an involution. When the data buffer is d bytes long and each byte is designated by $b(i)$, for i in the range $0 \leq i < d$: The first stage of CMC is as follows: Initialize a variable z to zero, For successive integer values of i in the range $0 \leq i < d$ form a variable q by: $q = z \oplus$ low order byte of i , where \oplus is the bitwise boolean Exclusive-OR operator, form variable k by: $k = \text{TBOX}[q]$, update $b(i)$ with: $b(i) = b(i) + k \bmod 256$, and update z with: $z = b(i) + z \bmod 256$. The second stage of CMC is involutory and comprises: for all values of i in the range $0 \leq i < (d-1)/2$:

$b(i) = b(i) \oplus (b(d-1-i) \text{ OR } 1)$, where OR is the bitwise boolean OR operator. CMC's final stage is the decryption that is inverse of the first stage: Initialize a variable z to zero, For successive integer values of i in the range $0 \leq i < d$ form a variable q by: $q = z \oplus$ low order byte of i , form variable k by: $k = \text{TBOX}[q]$, update z with: $z = b(i) + z \bmod 256$, update $b(i)$ with: $b(i) = b(i) - k \bmod 256$.

The three stage process employed to encrypt and decrypt selected control and data messages is illustrated in FIG. 9. In one preferred embodiment the first stage and the third stage are an autokey encryption and decryption, respectively. An autokey system is a time-varying system where the output of the system is used to affect the subsequent output of the system. For further reference regarding cryptography and autokey systems, see W. Diffie and M. E. Hellman, Privacy and Authentication: An Introduction to Cryptography, Proc. of the I.E.E.E., Vol. 67, No. 3, March 1979.

Mobile Unit Apparatus

FIG. 10 presents a block diagram of a mobile unit hardware. It comprises a control block 200 which includes the key pad of a cellular telephone, the handset and the unit's power control switch. Control block 200 is connected to processor 210 which controls the workings of the mobile unit, such as converting speech signals to digital representation, incorporating error correction codes, encrypting the digital speech signal, decrypting incoming speech signals, forming

and encrypting (as well as decrypting) various control messages, etc. Block 210 is coupled to block 220 which comprises the bulk of the circuitry associated with transmission and reception of signals. Blocks 200-220 are basically conventional blocks, performing the functions that are currently performed by commercial mobile telephone units (though the commercial units do not do encrypting and decrypting). To incorporate the authentication and encryption processes disclosed herein, the apparatus of FIG. 8 also includes a block 240 which comprises a number of registers coupled to processor 210, and a "personality" module 230 that is also coupled to processor 210. Module 230 may be part of the physical structure of a mobile telephone unit, or it may be a removable (and pluggable) module that is coupled to the mobile telephone unit through a socket interface. It may also be coupled to processor 210 through an electromagnetic path, or connection. Module 230 may be, for example, a "smart card".

Module 230 comprises a Jumble processor 231 and a number of registers associated with processor 231. Alternately, in another preferred embodiment, only the A-Key is in the module 230. A number of advantages accrue from installing (and maintaining) the A-key, and the MIN1 and MIN2 designations in the registers of module 230, rather than in the registers of block 240. It is also advantageous to store the developed SSD field in the registers of module 230. It is further advantageous include among the registers of module 230 any needed working registers for carrying out the processes of processor 231. By including these elements in module 230, the user may carry the module on his person to be used with different mobile units (e.g. "extension" mobile units) and more of the sensitive information is stored outside the module. Of course, mobile units may be produced with module 230 being an integral and permanent part of the unit. In such embodiments, Jumble processor 231 may be merged within processor 210. Block 240 stores the unit's ESN designation and the various RAND sequences that are received.

Although the above disclosure is couched in terms of subscriber authentication in a cellular telephony environment, and that includes personal communication networks which will serve portable wallet sized handsets, it is clear that the principles of this invention have applicability in other environments where the communication is perceived to be not sufficiently secure and where impersonation is a potential problem. This includes computer networks, for example.

Claims

1. In a communications system, a method of transforming a set of message signals representing a

message, the method which includes the steps of encrypting said set of message signals with an encryption process and a set of key signals to form a set of first intermediate signals,

altering said set of first intermediate signals in accordance with an involutory transformation to form a set of second intermediate signals, and

decrypting said set of second intermediate signals with a decryption process which is the inverse of said encryption process to form a set of output signals,

CHARACTERIZED IN THAT:

said step of altering comprises the step of modifying a first subset of said set of first intermediate signals with an unkeyed transformation based on a second subset of said set of first intermediate signals.

2. The method of claim 1 wherein said step of encrypting comprises the step of forming said set of first intermediate signals in accordance with a first autokey process and said step of decrypting comprises the step of forming said set of output signals in accordance with a second autokey process.
3. The method of claim 1 wherein said set of key signals has N elements and said set of message signals has D elements, where N and D are positive integers.
4. The method of claim 3 wherein said step of encrypting said set of message signals comprises the steps of:
 - setting a signal z to a selected magnitude; and
 - for different integral magnitudes of a signal i , spanning the range $0 \leq i < D$,
 - setting a signal q based on the respective magnitudes of said signals i and z ;
 - setting a signal k based on the respective magnitudes of said signal q and on a signal T , where T is the q^{th} element of said set of key signals;
 - creating an i^{th} element of said set of first intermediate signals based on the respective magnitudes of said i^{th} element of said set of message signals and said signal k ; and
 - updating the magnitude of said signal z based on the respective magnitudes of said i^{th} element of said set of first intermediate signals and said signal z .
5. The method of claim 3 wherein said step of encrypting said set of message signals comprises the steps of:
 - setting an n -tet signal z to a selected val-

ue; and

for successive integer values of index i , spanning the range $0 \leq i < D$,

setting an n -tet signal q to $z \oplus m$, where \oplus is the bit-wise boolean Exclusive-OR operator and m is i modulo 2^n ;

setting an n -tet signal k to T , where T is said q^{th} n -tet element of said set of key signals;

creating an i^{th} n -tet element of said set of first intermediate signals by adding, modulo 2^n , said i^{th} n -tet element of said set of message signals and said n -tet signal k ; and

updating the value of said n -tet signal z by adding, modulo 2^n , said i^{th} n -tet element of said set of first intermediate signals and said n -tet signal z .

6. The method of claim 3 wherein said step of altering said set of first intermediate signals creates, in the range $0 \leq i < \frac{(D-1)}{2}$, an i^{th} element of said set of second intermediate signals equal to $b(i) \oplus q$, where $b(i)$ is the i^{th} element of said set of first intermediate signals, where q is based on $b(x)$, where x is based on the values of D and i , where $b(x)$ is the x^{th} element of said set of first intermediate signals, and where \oplus is the bit-wise boolean Exclusive-OR operator.
7. The method of claim 3 wherein said step of decrypting said set of second intermediate signals comprises the steps of:
 - setting a signal z to a selected magnitude; and
 - for different integral magnitudes of a signal i , spanning the range $0 \leq i < D$,
 - setting a signal q based on the respective magnitudes of said signals i and z ;
 - setting a signal k based on the respective magnitudes of said signal q and on a signal T , where T is the q^{th} element of said set of key signals;
 - updating the magnitude of said signal z based on the respective magnitudes of said i^{th} element of said set of second intermediate signals and said signal z ; and
 - creating an i^{th} element of said set of output signals based on the respective magnitudes of said i^{th} element of said set of second intermediate signals and said signal k .
8. The method of claim 3 wherein said step of decrypting said set of second intermediate signals comprises the steps of:
 - setting an n -tet signal z to a selected magnitude; and
 - for successive integer values of index i

spanning the range $0 \leq i < D$,

setting an n -tet signal q to $z \oplus m$,
where \oplus is the bit-wise boolean Exclusive-OR
operator and m is i modulo 2^n ;

setting an n -tet signal k to T , where
 T is said q^{th} n -tet element of said set of key sig-
nals;

updating the value of said n -tet sig-
nal z by adding, modulo 2^n , said i^{th} n -tet element
of said set of second intermediate signals and
said signal z ; and

creating an i^{th} n -tet element of said
set of output signals by subtracting, modulo 2^n , n -
tet signal k from the i^{th} element of said set of sec-
ond intermediate signals.

9. The method of claim 1 wherein said set of mes-
sage signals represents a message in a cellular
telephone system.

10. The method of claim 1 wherein said set of mes-
sage signals represent speech.

11. The method of claim 1 wherein said set of mes-
sage signals represent non-speech data.

12. A cryptographic system for transforming a set of
message signals which set represents a mes-
sage comprising:

means for encrypting said set of
message signals with an encryption process and
a set of key signals to form a set of first intermedi-
ate signals;

means for altering said set of first
intermediate signals in accordance with an invol-
utory transformation to form a set of second in-
termediate signals; and

means for decrypting said set of
second intermediate signals with a decryption
process which is the inverse of said encryption
process to form a set of output signals,

CHARACTERIZED IN THAT:

said means for altering comprises means
for modifying a first subset of said set of first in-
termediate signals with an unkeyed transforma-
tion in accordance with a second subset of said
set of first intermediate signals.

13. The communications system of claim 12 wherein
said means for encrypting comprises means for
forming said set of first intermediate signals in
accordance with a first autokey process and said
means for decrypting comprises means for form-
ing said set of output signals in accordance with
a second autokey process.

14. The communications system of claim 12 wherein
said set of key signals has N elements and said

set of message signals has D elements, where N
and D are positive integers.

15. The communications system of claim 14 wherein
said means for encrypting said set of message
signals comprises:

means for setting a signal z to a selected
magnitude; and

means for setting a signal i to different in-
tegral magnitudes spanning the range $0 \leq i < D$,

means for setting a signal q based on the
respective magnitudes of said signals i and z ;

means for setting a signal k based on the
respective magnitudes of said signal q and on a
signal $K(q)$, where $K(q)$ is the q^{th} element of said
set of key signals;

means for creating an i^{th} element of said
set of first intermediate signals based on the re-
spective magnitudes of said i^{th} element of said set
of message signals and said signal k ; and

means for updating the magnitude of said
signal z based on the respective magnitudes of
said i^{th} element of said set of first intermediate
signals and said signal z .

16. The communications system of claim 14 wherein
said means for altering said set of first intermedi-
ate signals creates, in the range $0 \leq i < \frac{(D-1)}{2}$,

an i^{th} element of said set of second intermediate
signals equal to $b(i) \oplus q$, where $b(i)$ is the i^{th} ele-
ment of said set of first intermediate signals,
where q is based on $b(x)$, where x is based on the
values of D and i , where $b(x)$ is the x^{th} element of
said set of first intermediate signals, and where
 \oplus is the bit-wise boolean Exclusive-OR operator.

17. The communications system of claim 14 wherein
said means for decrypting said set of second in-
termediate signals comprises:

means for setting a signal z to a selected
magnitude; and

means for setting a signal i to different in-
tegral magnitudes spanning the range $0 \leq i < D$,

means for setting a signal q based on the
respective magnitudes of said signals i and z ;

means for setting a signal k based on the
respective magnitudes of said signal q and on a
signal $K(q)$, where $K(q)$ is the q^{th} element of said
set of key signals;

means for updating the magnitude of said
signal z based on the respective magnitudes of
said i^{th} element of said set of second intermediate
signals and said signal z ; and

means for creating an i^{th} element of said
set of output signals based on the respective
magnitudes of said i^{th} element of said set of sec-
ond intermediate signals and said signal k .

FIG. 1

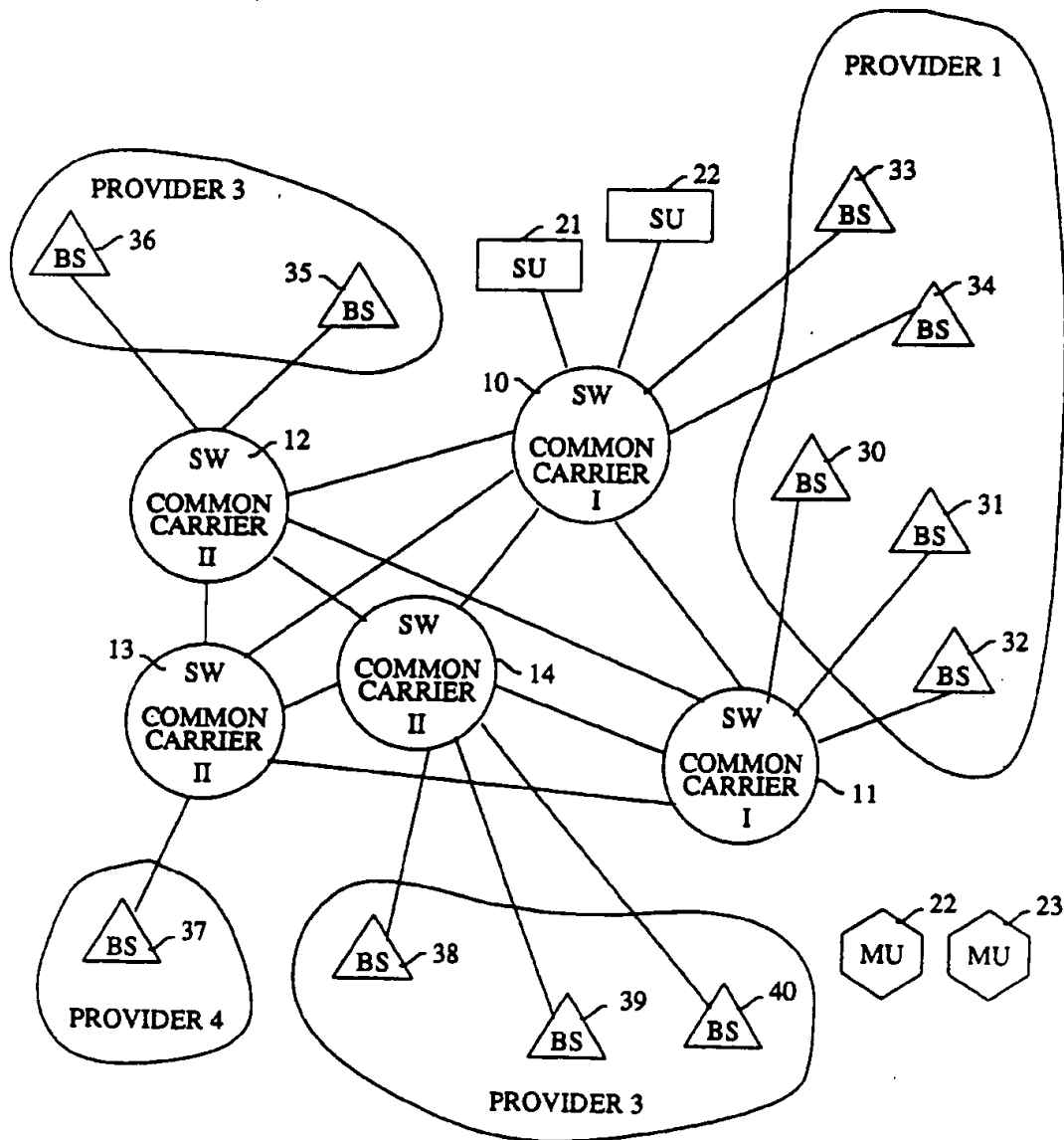


FIG. 2

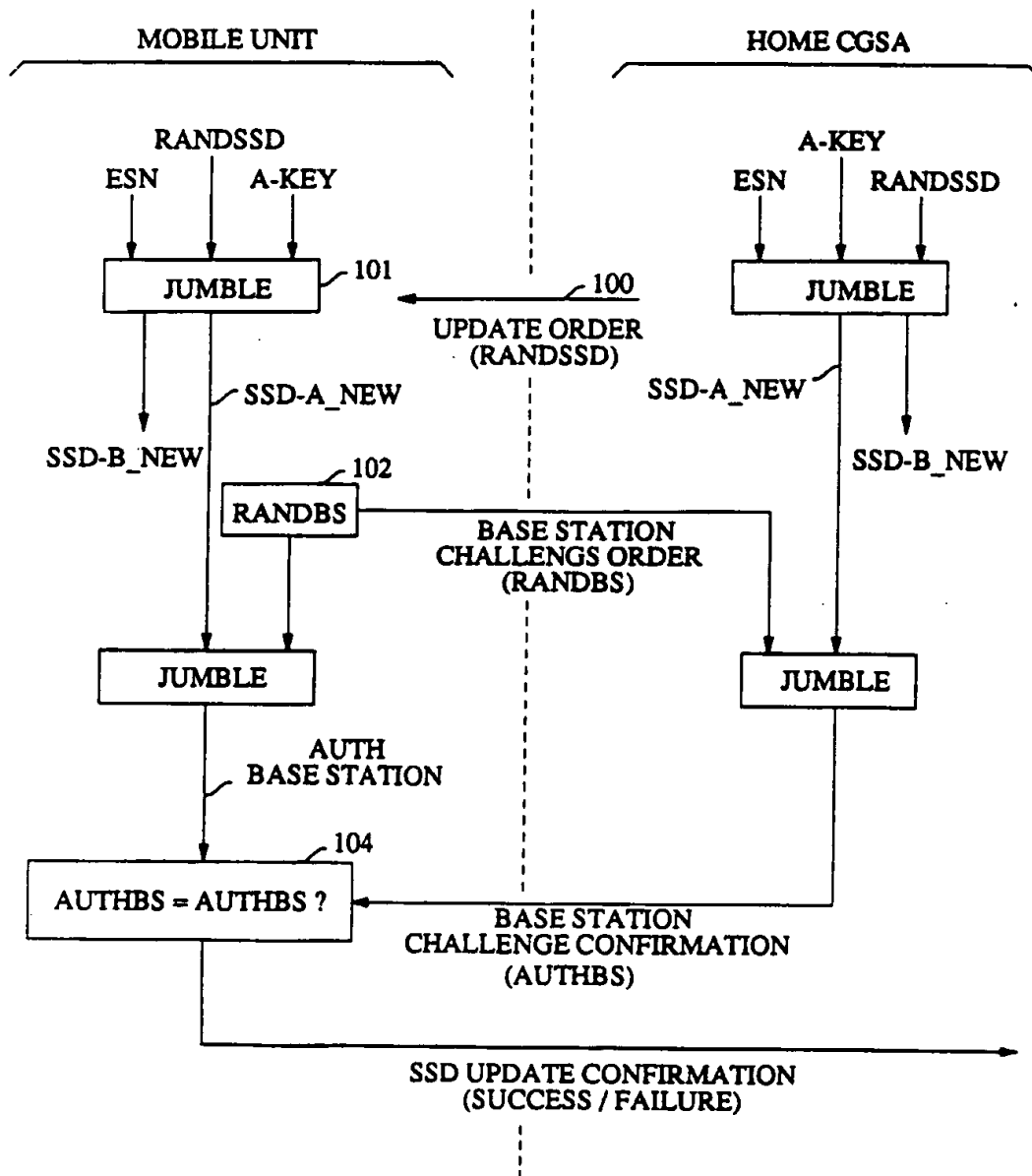


FIG. 3

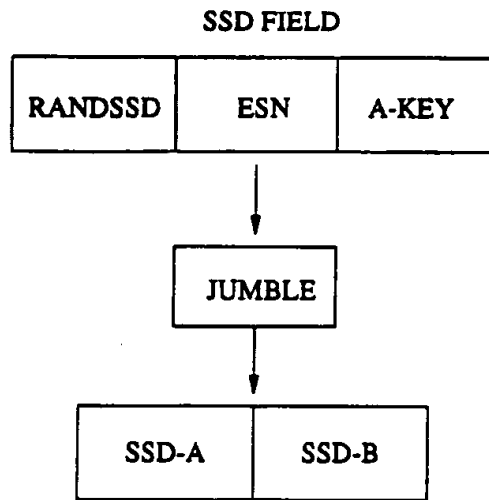


FIG. 4

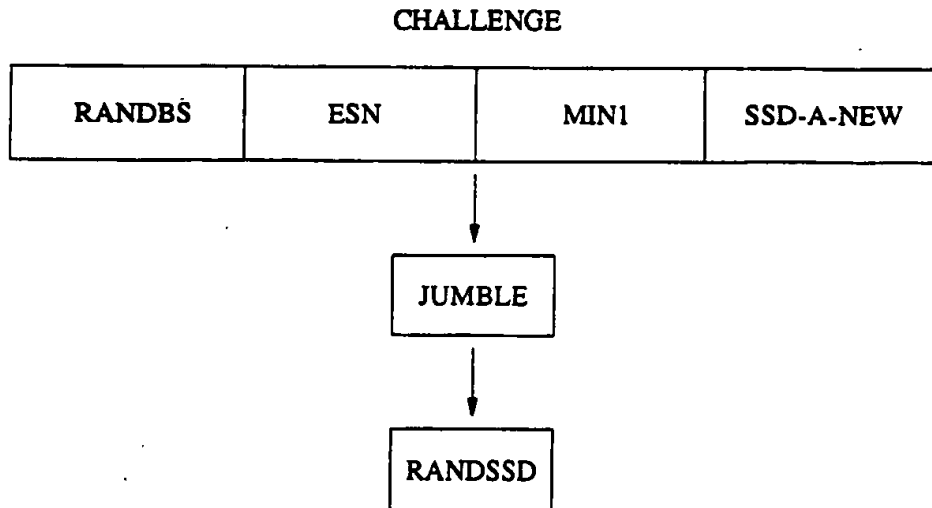


FIG. 5

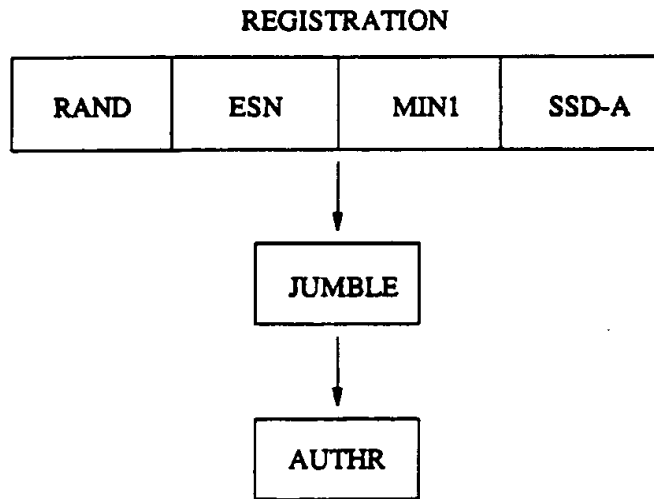


FIG. 6

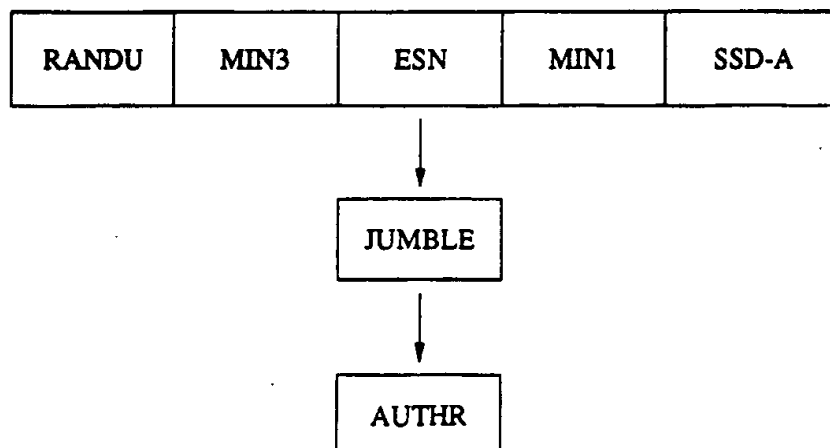


FIG. 7

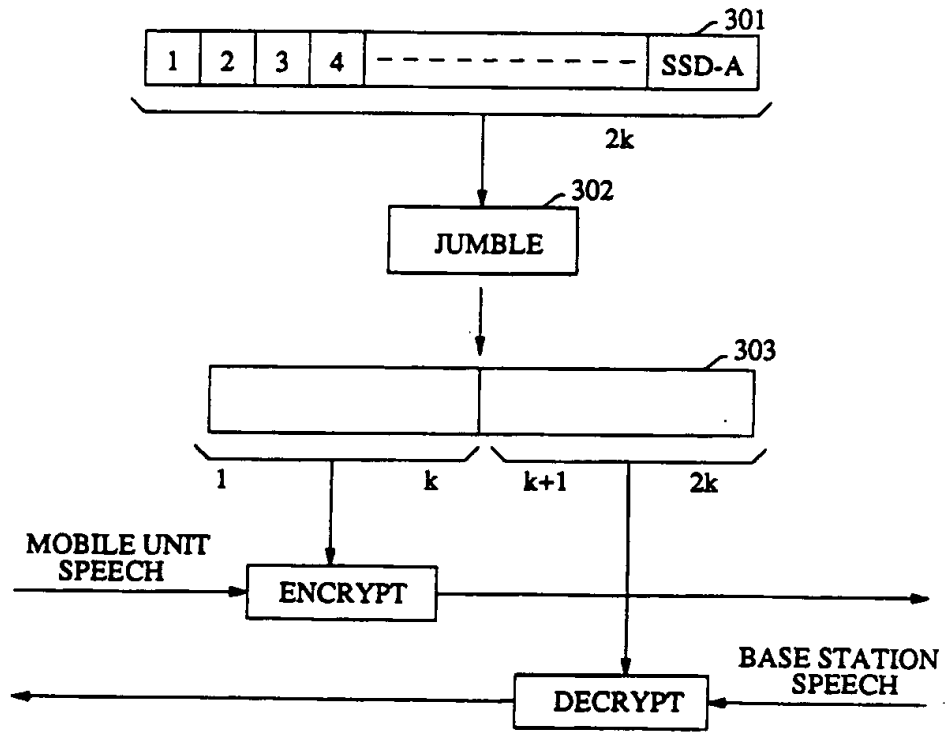


FIG. 8

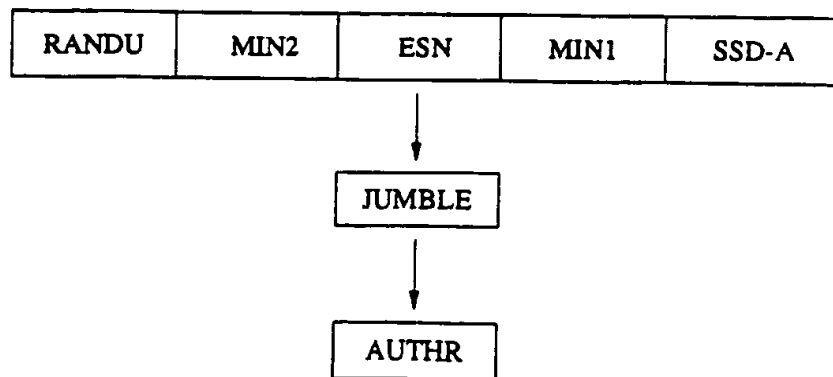


FIG. 9

CONTROL MESSAGE CRYPTOSYSTEM

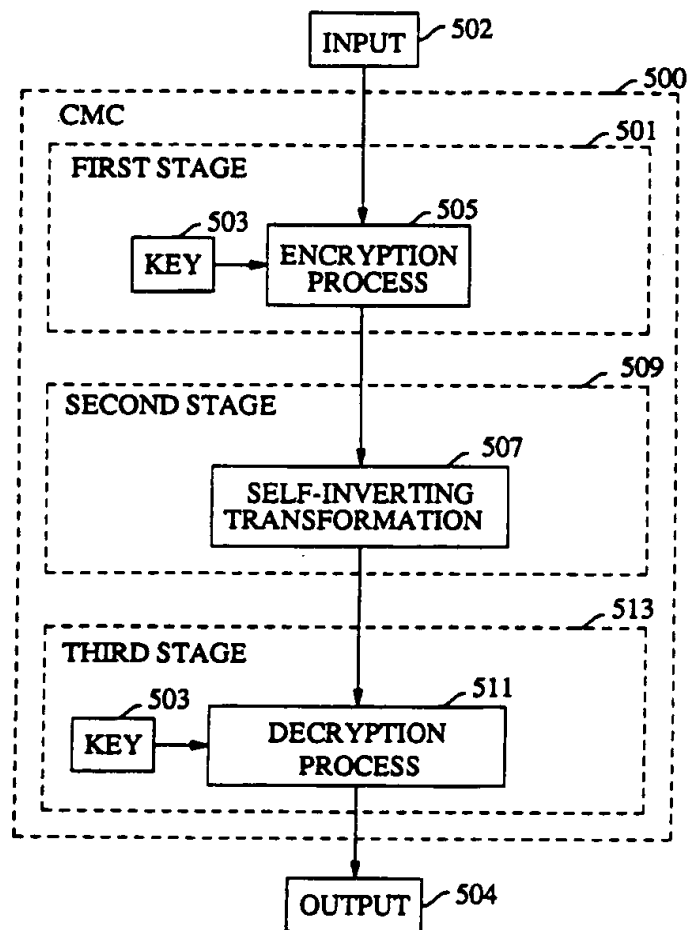


FIG. 10

